Requested Patent: WO9900936A1

Title: HIGHLY INTEGRATED MULTI-LAYER SWITCH ELEMENT ARCHITECTURE ;

Abstracted Patent: US6246680 ;

Publication Date: 2001-06-12 ;

Inventor(s):

MULLER SHIMON (US); FRAZIER HOWARD (US); HENDEL ARIEL (US) ;

Applicant(s): SUN MICROSYSTEMS INC (US) ;

Application Number: US19970884704 19970630 ;

Priority Number(s): US19970884704 19970630 ;

IPC Classification: H04L12/56 ;

Equivalents: EP1002397 (WO9900936), A4

ABSTRACT:

An architecture for a highly integrated network element building block is provided. According to one aspect of the present invention, a network device building block includes a network interface with multiple ports for transmitting and receiving packets over a network. The network device building block also includes a packet buffer storage which is coupled to the network interface. The packet buffer storage acts as an elasticity buffer for adapting between incoming and outgoing bandwidth requirements. A shared memory manager may also be provided dynamically allocate and deallocate buffers in the packet buffer storage on behalf of the network interface and other clients of the packet buffer storage. The network device building block further includes a switch fabric which is coupled to the network interface. The switch fabric provides forwarding decisions for received packets. A given forwarding decision includes a list of ports upon which a particular received packet is to be forwarded. A central processing unit (CPU) interface is also included in the network device building block. The CPU interface is coupled to the switch fabric and is configured to forward packets received from the CPU based upon forwarding decisions provided by the switch fabric

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 12/28 — A1

(11) International Publication Number: WO 99/00936

(43) International Publication Date: 7 January 1999 (07.01.99)

(21) International Application Number: PCT/US98/13199

(22) International Filing Date: 24 June 1998 (24.06.98)

(30) Priority Data:
08/884,704 30 June 1997 (30.06.97) US

(71) Applicant: SUN MICROSYSTEMS, INC. [US/US]; 901 San Antonio Road, Palo Alto, CA 94303 (US).

(72) Inventors: MULLER, Shimon; 983 La Mesa Tr., Sunnyvale, CA 94086 (US). HENDEL, Ariel; 7537 Newcastle Drive, Cupertino, CA 95014 (US). FRAZIER, Howard; 4951 Middleton Place, Pleasonton, CA 94566 (US).

(74) Agents: HYMAN, Eric, S. et al.; Blakely, Sokoloff, Taylor & Zafman, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025-1026 (US).

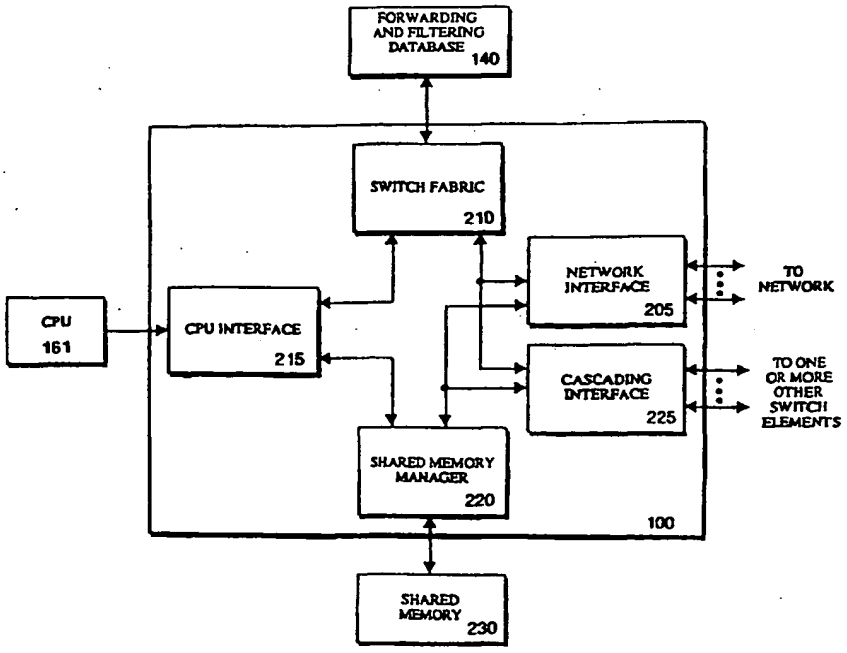
(81) Designated States: JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: A HIGHLY INTEGRATED MULTI-LAYER SWITCH ELEMENT ARCHITECTURE

(57) Abstract

An architecture for a highly integrated network element building block (100) is provided. According to one aspect of the present invention, a network device building block (100) includes a network interface (205) with multiple ports for transmitting and receiving packets over a network. The network device building block (100) also includes a packet buffer storage (230) which is coupled to the network interface. The packet buffer storage (230) acts as an elasticity buffer for adapting between incoming and outgoing bandwidth requirements. A shared memory manager (220) may also be provided dynamically to allocate and deallocate buffers in the packet buffer storage (230) on behalf of the network interface (205) and other clients of the packet buffer storage (230). The network device building block (100) furhter includes a switch fabric (210) which is coupled to the network interface (205). The switch fabric (210) provides forwarding decisions for received packets. A given forwarding decision includes a list of ports upon which a particular received packet is to be forwarded. A central processing unit (CPU) interface (215) is also included in the network device building block (100). The CPU interface (215) is coupled to the switch fabric (210) and is configured to forward packets received from the CPU (161) based upon forwarding decisions provided by the switch fabric (210).



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

A HIGHLY INTEGRATED MULTI-LAYER SWITCH ELEMENT ARCHITECTURE

FIELD OF THE INVENTION

The invention relates generally to the field of computer networking devices. More particularly, the invention relates to an architecture for a highly integrated network element building block.

BACKGROUND OF THE INVENTION

An increasing number of users are requiring increased bandwidth from existing networks due to multimedia applications for accessing the Internet and World Wide Web, for example. Therefore, future networks must be able to support a very high bandwidth and a large number of users. Furthermore, such networks should be able to support multiple traffic types such as data, voice, and video which typically require different bandwidths.

Statistical studies indicate that the network domain, i.e., a group of interconnected local area networks (LANs), as well as the number of individual end-stations connected to each LAN, will grow at ever increasing rates in the future. Thus, more network bandwidth and more efficient use of resources is needed to meet these requirements.

Building networks using Layer 2 elements such as bridges provides fast packet forwarding between LANs; however there is no flexibility in traffic isolation, redundant topologies, and end-to-end policies for queuing and access control. While the latter attributes may be met using Layer 3 elements such as routers, packet forwarding speed is

sacrificed in return for the greater intelligence and decision making capabilities provided by routers.

Therefore, it is desirable to provide a cost-effective, high performance network device building block that is capable of performing non-blocking wire-speed multi-layer switching on N ports. Generally, it would be advantageous to provide a network device building block that linearly scales its performance with advances in silicon technology. Therefore, it is desirable to share common resources, centralize common processing, and maximize the utilization of hardware resources. More specifically, it is desirable to utilize a dynamic packet memory management scheme to facilitate sharing of a common packet memory among all input/output ports for packet buffering. Also, it is desirable to centralize packet header processing and to provide efficient access to a centralized database for multiple protocol layer based forwarding decisions. Further, it would be advantageous to provide a central processing unit (CPU) interface that requests forwarding decisions of a switch fabric for CPU originated packets in a first packet forwarding mode and bypasses the switch fabric header matching by transferring the packet directly to one or more specified ports in a second packet forwarding mode.

SUMMARY OF THE INVENTION

A method and apparatus for packet forwarding and filtering is described in the context of an architecture for a highly integrated network element building block. According to one aspect of the present invention, a network device building block includes a network interface with multiple ports for transmitting and receiving packets over a network. The network device building block also includes a packet buffer storage which is coupled to the network interface. The packet buffer storage acts as an elasticity buffer for adapting between incoming and outgoing bandwidth requirements. The network device building block further includes a switch fabric which is coupled to the network interface. The switch fabric provides forwarding decisions for received packets. A given forwarding decision includes a list of ports upon which a particular received packet is to be forwarded. A central processing unit (CPU) interface is also included in the network device building block. The CPU interface is coupled to the switch fabric and is configured to forward packets received from the CPU based upon forwarding decisions provided by the switch fabric.

According to another aspect of the present invention, a switch element includes a switch fabric configured to generate forwarding decisions for received packets. The switch element also includes multiple interfaces for receiving and transmitting packets. Each of the interfaces are coupled in communication with the switch fabric for requesting and receiving forwarding decisions. The interfaces include a network interface, a cascading interface, and a central processing unit (CPU) interface. The network interface further includes multiple external ports for communication with devices on a network. At least two internal links are provided by the cascading interface for interconnecting with one or more other switch elements in a full-mesh topology. The CPU interface allows communication of

packets and commands between the switch fabric and a CPU. The switch element further includes a shared memory manager which is coupled to the interfaces for dynamically allocating and deallocating buffers in a shared buffer memory on behalf of the interfaces. The shared memory manager further tracks the status of buffers in the shared buffer memory.

Other features of the present invention will be apparent from the accompanying drawings and from the detailed description which follows.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

Figure 1 illustrates a switch according to one embodiment of the present invention.

Figure 2 is a simplified block diagram of an exemplary switch element that may be utilized in the switch of Figure 1.

Figure 3 is a more detailed block diagram of the switch element of Figure 2.

DETAILED DESCRIPTION

A highly integrated multi-layer switch element architecture is described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without some of these specific details. In other instances, well-known structures and devices are shown in block diagram form.

The present invention includes various steps, which will be described below. While the steps of the present invention are preferably performed by the hardware components described below, alternatively, the steps may be embodied in machine-executable instructions, which may be used to cause a general-purpose or special-purpose processor programmed with the instructions to perform the steps. Further, embodiments of the present invention will be described with respect to a high speed Ethernet switch. However, it will be appreciated that the method and apparatus described herein are equally applicable to other types of network devices such as bridges, routers, brouters, and other network devices.

AN EXEMPLARY NETWORK ELEMENT

An overview of one embodiment of a network element that operates in accordance with the teachings of the present invention is illustrated in Figure 1. The network element is used to interconnect a number of nodes and end-stations in a variety of different ways. In particular, an application of the multi-layer distributed network element (MLDNE) would be to route packets according to predefined routing protocols over a homogenous data link

layer such as the IEEE 802.3 standard, also known as the Ethernet. Other routing protocols can also be used.

The MLDNE's distributed architecture can be configured to route message traffic in accordance with a number of known or future routing algorithms. In a preferred embodiment, the MLDNE is configured to handle message traffic using the Internet suite of protocols, and more specifically the Transmission Control Protocol (TCP) and the Internet Protocol (IP) over the Ethernet LAN standard and medium access control (MAC) data link layer. The TCP is also referred to here as a Layer 4 protocol, while the IP is referred to repeatedly as a Layer 3 protocol.

In one embodiment of the MLDNE, a network element is configured to implement packet routing functions in a distributed manner, i.e., different parts of a function are performed by different subsystems in the MLDNE, while the final result of the functions remains transparent to the external nodes and end-stations. As will be appreciated from the discussion below and the diagram in Figure 1, the MLDNE has a scalable architecture which allows the designer to predictably increase the number of external connections by adding additional subsystems, thereby allowing greater flexibility in defining the MLDNE as a stand alone router.

As illustrated in block diagram form in Figure 1, the MLDNE 101 contains a number of subsystems 110 that are fully meshed and interconnected using a number of internal links 141 to create a larger switch. At least one internal link couples any two subsystems. Each subsystem 110 includes a switch element 100 coupled to a forwarding and filtering database 140, also referred to as a forwarding database. The forwarding and filtering database may include a forwarding memory 113 and an associated memory 114. The forwarding memory (or database) 113 stores an address table used for matching with

the headers of received packets. The associated memory (or database) stores data associated with each entry in the forwarding memory that is used to identify forwarding attributes for forwarding the packets through the MLDNE. A number of external ports (not shown) having input and output capability interface the external connections 117. In one embodiment, each subsystem supports multiple Gigabit Ethernet ports, Fast Ethernet ports and Ethernet ports. Internal ports (not shown) also having input and output capability in each subsystem couple the internal links 141. Using the internal links, the MLDNE can connect multiple switching elements together to form a multigigabit switch.

The MLDNE 101 further includes a central processing system (CPS) 160 that is coupled to the individual subsystem 110 through a communication bus 151 such as the peripheral components interconnect (PCI). The CPS 160 includes a central processing unit (CPU) 161 coupled to a central memory 163. Central memory 163 includes a copy of the entries contained in the individual forwarding memories 113 of the various subsystems. The CPS has a direct control and communication interface to each subsystem 110 and provides some centralized communication and control between switch elements.

AN EXEMPLARY SWITCH ELEMENT

Figure 2 is a simplified block diagram illustrating an exemplary architecture of the switch element of Figure 1. The switch element 100 depicted includes a central processing unit (CPU) interface 215, a switch fabric block 210, a network interface 205, a cascading interface 225, and a shared memory manager 220.

Ethernet packets may enter or leave the network switch element 100 through any one of the three interfaces 205, 215, or 225. In brief, the network interface 205 operates in accordance with a corresponding Ethernet protocol to receive Ethernet packets from a

network (not shown) and to transmit Ethernet packets onto the network via one or more external ports (not shown). An optional cascading interface 225 may include one or more internal links (not shown) for interconnecting switching elements to create larger switches. For example, each switch element 100 may be connected together with other switch elements in a full mesh topology to form a multi-layer switch as described above. Alternatively, a switch may comprise a single switch element 100 with or without the cascading interface 225.

The CPU 161 may transmit commands or packets to the network switch element 100 via the CPU interface 215. In this manner, one or more software processes running on the CPU 161 may manage entries in an external forwarding and filtering database 140, such as adding new entries and invalidating unwanted entries. In alternative embodiments, however, the CPU 161 may be provided with direct access to the forwarding and filtering database 140. In any event, for purposes of packet forwarding, the CPU port of the CPU interface 215 resembles a generic input port into the switch element 100 and may be treated as if it were simply another external network interface port. However, since access to the CPU port occurs over a bus such as a peripheral components interconnect (PCI) bus, the CPU port does not need any media access control (MAC) functionality.

Returning to the network interface 205, the two main tasks of input packet processing and output packet processing will now briefly be described. Input packet processing may be performed by one or more input ports of the network interface 205. Input packet processing includes the following: (1) receiving and verifying incoming Ethernet packets, (2) modifying packet headers when appropriate, (3) requesting buffer pointers from the shared memory manager 220 for storage of incoming packets, (4) requesting forwarding decisions from the switch fabric block 210, (5) transferring the

incoming packet data to the shared memory manager 220 for temporary storage in an external shared memory 230, and (5) upon receipt of a forwarding decision, forwarding the buffer pointer(s) to the output port(s) indicated by the forwarding decision. Output packet processing may be performed by one or more output ports of the network interface 205. Output processing includes requesting packet data from the shared memory manager 220, transmitting packets onto the network, and requesting deallocation of buffer(s) after packets have been transmitted.

The network interface 205, the CPU interface 215, and the cascading interface 225 are coupled to the shared memory manager 220 and the switch fabric block 210. Preferably, critical functions such as packet forwarding and packet buffering are centralized as shown in Figure 2. The shared memory manager 220 provides an efficient centralized interface to the external shared memory 230 for buffering of incoming packets. The switch fabric block 210 includes a search engine and learning logic for searching and maintaining the forwarding and filtering database 140 with the assistance of the CPU 161.

The centralized switch fabric block 210 includes a search engine that provides access to the forwarding and filtering database 140 on behalf of the interfaces 205, 215, and 225. Packet header matching, Layer 2 based learning, Layer 2 and Layer 3 packet forwarding, filtering, and aging are exemplary functions that may be performed by the switch fabric block 210. Each input port is coupled with the switch fabric block 210 to receive forwarding decisions for received packets. The forwarding decision indicates the outbound port(s) (e.g., external network port or internal cascading port) upon which the corresponding packet should be transmitted. Additional information may also be included in the forwarding decision to support hardware routing such as a new MAC destination address (DA) for MAC DA replacement. Further, a priority indication may also be

included in the forwarding decision to facilitate prioritization of packet traffic through the switch element 100.

In the present embodiment, Ethernet packets are centrally buffered and managed by the shared memory manager 220. The shared memory manager 220 interfaces every input port and output port and performs dynamic memory allocation and deallocation on their behalf, respectively. During input packet processing, one or more buffers are allocated in the external shared memory 230 and an incoming packet is stored by the shared memory manager 220 responsive to commands received from the network interface 205, for example. Subsequently, during output packet processing, the shared memory manager 220 retrieves the packet from the external shared memory 230 and deallocates buffers that are no longer in use. To assure no buffers are released until all output ports have completed transmission of the data stored therein, the shared memory manager 220 preferably also tracks buffer ownership.

Having described the architecture of the switch element 100 at a high level, a more detailed view of the individual components will now be described with reference to Figure 3.

NETWORK AND CASCADING INTERFACES

The switch element of the present invention provides wire speed routing and forwarding of Ethernet, Fast Ethernet, and Gigabit Ethernet packets among the three interfaces 215, 205, and 225. According to the present embodiment, each port of the network interface 205 and the cascading interface includes input packet process (IPP), an output packet process (OPP), and a media access controller (MAC).

The IPPs are coupled in communication with the switch fabric 210, the shared memory manager 220, and the OPPs. The IPPs request forwarding decisions from the switch fabric 210 for received packets and temporarily store the packet data in the shared memory 230 until a forwarding decision is returned. Upon receipt of a forwarding decision, the IPPs forward the corresponding packet to the appropriate OPPs, if any.

According to one embodiment, received packet headers are modified by the IPPs as disclosed in U.S. Patent Application Number _____, entitled "Mechanism for Packet Field Replacement in a Multi-Layered Switched Network Element" filed on June 30, 1997, attorney docket number 082225.P2376 which is incorporated herein by reference.

The OPPs are coupled in communication with the shared memory manager 220. When a packet is ready for transmission, the OPPs retrieve the packet data from the shared memory 230 via the shared memory manager 220 and transmit the packet data onto the attached network.

According to one embodiment, dynamic output queuing in the OPPs is as disclosed in U.S. Patent Application Number _____, entitled "Method and Apparatus for Dynamic Queue Sizing" filed on June 30, 1997, attorney docket number 082225.P2377 which is incorporated herein by reference.

According to another embodiment, packet routing and packet field replacement are as disclosed in U.S. Patent Application Number _____, entitled "Mechanism for Packet Field Replacement in a distributed Multi-Layer Network Element" filed on June 30, 1997, attorney docket number 082225.P2583 which is incorporated herein by reference.

SWITCH FABRIC

The switch fabric 210 provides centralized access to the forwarding and filtering database 140 on behalf of the input ports. Highly pipelined logic within the switch fabric 210 allows it to receive and process packet headers from several input ports at once. Advantageously, the centralization and pipelining reduce hardware implementation overhead. For example, an N stage packet header processing pipeline allows N packet headers to be processed from various input ports in a single block rather than having to provide N individual packet header processing units.

According to one embodiment, the switch fabric 210 is implemented as disclosed in U.S. Patent Application Number _____, entitled "Search Engine Architecture for a High Performance Multi-Layer Switch Element" filed on June 30, 1997, attorney docket number 082225.P2361 and U.S. Patent Application Number _____, entitled "Hardware-Assisted Central Processing Unit Access to a Forwarding Database" filed on June 30, 1997, attorney docket number 082225.P2559, the contents of which are incorporated herein by reference.

CPU INTERFACE

The CPU interface 215 of the present embodiment, includes a single CPU port comprising a bus interface (BIF) 340 coupled to a host transmit process (HTP) 350 and a host receive process (HRP) 360. The BIF 340 implements a bus interface protocol for communicating data between the CPU 161 and the switch element 100. In this respect, it has similar responsibilities as the MACs in the network interface ports. In one embodiment, the BIF 340 includes a PCI protocol block for supporting PCI direct memory accesses (DMAs) to and from the CPU memory.

Architecturally the CPU port is designed to mirror the network interface ports and cascading interface ports. The BIF 340 corresponds functionally with a network interface port MAC. For example, both the BIF 340 and the MACs are responsible for dealing with a particular protocol (e.g., PCI and Ethernet, respectively) for communication over a physical medium with devices external to the switch element 100. Similarly, the HTP 350 corresponds to an IPP of a network interface port. Both the HTP 350 and the IPP are responsible for buffering incoming packets, requesting forwarding decisions from the switch fabric 210, and transferring incoming packets to appropriate output port(s). Finally, the HRP 360 corresponds to an OPP of a network interface port. Both the HRP 360 and the OPP are responsible for retrieving outbound packets from the shared memory 230, transmitting outbound packets, and notifying the shared memory manager 220 when packet buffer pointers may be released. This novel CPU interface architecture allows the CPU port to be treated by the other switch element components as if it were simply another network interface port.

Another feature of the CPU interface 215 is the availability of two forwarding modes for CPU originated packets. In one embodiment, the CPU interface 215 is configured to operate in one of two forwarding modes with respect to a given packet based upon per packet control information provided by the CPU. In the first mode, the switch mode, the CPU interface 215 requests forwarding decisions from the switch fabric 210 and in the second mode, the directed mode, switch fabric header matching is bypassed, thereby allowing the packet to be directly transferred to one or more specified ports.

Since the switch fabric 210 and the shared memory manager 220 generally interact with the CPU interface 215 as if it were another network interface, the switch mode for CPU originated packets parallels the packet forwarding that takes place at the network

interface 205 and cascading interface 225. As will be appreciated, the innovative design and treatment of the CPU interface 215 provide for an efficient implementation of the novel switch mode for CPU originated packets.

Referring now to the directed mode, the CPU interface 215 forwards CPU originated packets based on control information that accompanies the CPU originated packets. The control information may contain information about the packet for facilitating packet processing by the switch element 100. For example, in one embodiment, a directed mode flag may be provided within the control information to indicate that the packet is to be transferred to one or more specified output ports rather than forwarded with reference to a forwarding decision provided by the switch fabric 210. In this case, the typical packet header matching and forwarding database search will be bypassed, and the packet will be transferred to the specified output port(s). It is appreciated that other flags and control information may also be incorporated into the control information.

SHARED MEMORY MANAGER

According to the present embodiment, the shared memory manager 220 includes a buffered architecture that utilizes a shared pool of packet memory and a dynamic buffer allocation scheme. Prior input port buffering and output buffering packet buffering schemes typically have a static portion of memory associated with each port, resulting in inefficient memory allocation and buffering that is not related to the actual amount of traffic through a given port. In contrast, the memory management provided by the present invention is designed to achieve efficient allocation of per port buffering that is proportional to the amount of traffic through a given port.

The shared memory manager 220 provides an efficient centralized interface to the shared memory 230 for buffering of incoming packets. The shared memory 230 is a pool of buffers that are used for temporary storage of packet data en route from an inbound interface (e.g., IPP 310-314 or HTP 350) to one or more outbound interfaces (e.g., OPP 315-319 or HTP 360). Essentially, the shared memory serves as an elasticity buffer for adapting between the incoming and outgoing bandwidth requirements.

According to this embodiment, the shared memory manager 220 includes a buffer manager 325. A level of indirection is provided by the buffer manager 325 which is exploited by the input and output ports by queuing packet pointers instead of the packet data itself. As such, the buffering provided by the present invention does not fit into the prior buffering categories such as input packet buffering or output packet buffering. Rather, the buffering described herein is best described as shared memory buffering with output queuing. Advantageously, since pointers are queued at the ports, the act of switching, according to the present embodiment, is reduced to transferring a packet pointer between an input port to a specific queue of one or more output ports.

Each buffer in the shared memory 230 may be owned by one or more different ports at different points in time. For example, copies of a multicast packet's buffer pointer(s) may reside in several output port queues. In the embodiment depicted, the shared memory manager 220 also includes a pointer random access memory (PRAM) 320 coupled to the buffer manager 325. The pointer RAM 320 is an on-chip pointer table that stores usage counts for pages (buffers) of the shared memory 230. In this manner, the number of buffer owners at a given time is known by the buffer manager 325. Thus allowing the buffer manager 325 to perform dynamic deallocation of buffers upon release by the last output port.

According to one embodiment, the buffer manager 325 is implemented as disclosed in U.S. Patent Application Number _____, entitled "Hardware-Assisted Central Processing Unit Access to a Forwarding Database" filed on June 30, 1997, attorney docket number 082225.P2354, the contents of which is incorporated herein by reference.

Buffer memory controller 330 provides a centralized interface to the input and output ports for storing and retrieving packet data, respectively, to the shared memory 230. According to one embodiment, the buffer memory controller 330 is implemented as disclosed in U.S. Patent Application Number _____, entitled "Method and Apparatus For Arbitrating Access to a Shared Memory by Network Ports Operating at Different Data Rates" filed on June 30, 1997, attorney docket number 082225.P2501 and U.S. Patent Application Number _____, entitled "Method and Apparatus In a Packet Routing Switch for Controlling Access at Different Data Rates to a Shared Memory", filed on June 30, 1997, attorney docket number 082225.P2367, the contents of which are incorporated herein by reference.

Thus, a buffered architecture has been described which provides temporary storage of received packets in a shared pool of packet memory and provides for efficient allocation of per port buffering that is proportional to the amount of traffic through a given port

In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A network device building block comprising:
a network interface including a plurality of ports for transmitting and receiving packets over a network;
packet buffer storage coupled to the network interface acting as an elasticity buffer for adapting between incoming and outgoing bandwidth requirements;
a switch fabric coupled to the network interface for providing a forwarding decision corresponding to a received packet, the forwarding decision including a list of ports upon which the received packet is to be forwarded; and
a central processing unit (CPU) interface coupled to the switch fabric, the CPU interface configured to forward packets received from a CPU based upon forwarding decisions provided by the switch fabric.

2. The network device building block of claim 1, further comprising a cascading interface for coupling the network device building block to one or more other network device building blocks to form a larger switching device;

3. A switch element comprising:
a switch fabric configured to generate forwarding decisions for received packets;
a plurality of interfaces for receiving and transmitting packets, the plurality of interfaces coupled to the switch fabric for requesting and receiving forwarding decisions, the plurality of interfaces including
a network interface providing a plurality of external ports for communication with devices on a network,

a cascading interface providing at least two internal links for interconnecting with one or more other switch elements in a full-mesh topology, and

a central processing unit (CPU) interface for communication of packets and commands between the switch fabric and a CPU; and

a shared memory manager coupled to the plurality of interfaces for dynamically allocating and deallocating buffers in a shared buffer memory on behalf of the plurality of interfaces and tracking the status of buffers in the shared buffer memory.

4. A method of forwarding a packet onto a network of devices, the method comprising the steps of:

a central processing unit generating a packet for transmission onto a network of devices;

a switch element receiving the packet on a central processing unit (CPU) interface of a switching element; and

if the packet is associated with a first mode, then

the CPU interface requesting a forwarding decision for the packet from a switch fabric,

the CPU interface receiving the forwarding decision from the switch fabric,

the CPU interface forwarding the packet to the one or more ports indicated by the forwarding decision, and

the one or more ports transmitting the packet onto the network of devices.

5. The method of claim 4, wherein the packet is associated with control information, the method further comprising the steps of:

based upon the control information, the CPU interface determining that the packet is associated with a second mode; and

the CPU interface forwarding the packet to a port indicated in the control information;

6. A network device building block comprising:

a plurality of ports for receiving and transmitting packets over a network segment;

a shared memory manager coupled to each of the plurality of ports, the shared memory manager configured to dynamically allocate buffers from a shared memory for temporary storage of incoming packets, the shared memory manager further configured to release buffers associated with a particular packet when all ports to which the packet was forwarded have completed transmission of the packet.

a switch fabric coupled to each of the plurality of ports to provide a centralized interface to a forwarding and filtering database associated with the switch fabric, the forwarding and filtering database having stored therein forwarding information for a plurality of protocol layers, the switch fabric configured to perform header matching and searching of the forwarding and filtering database on behalf of the plurality of ports.

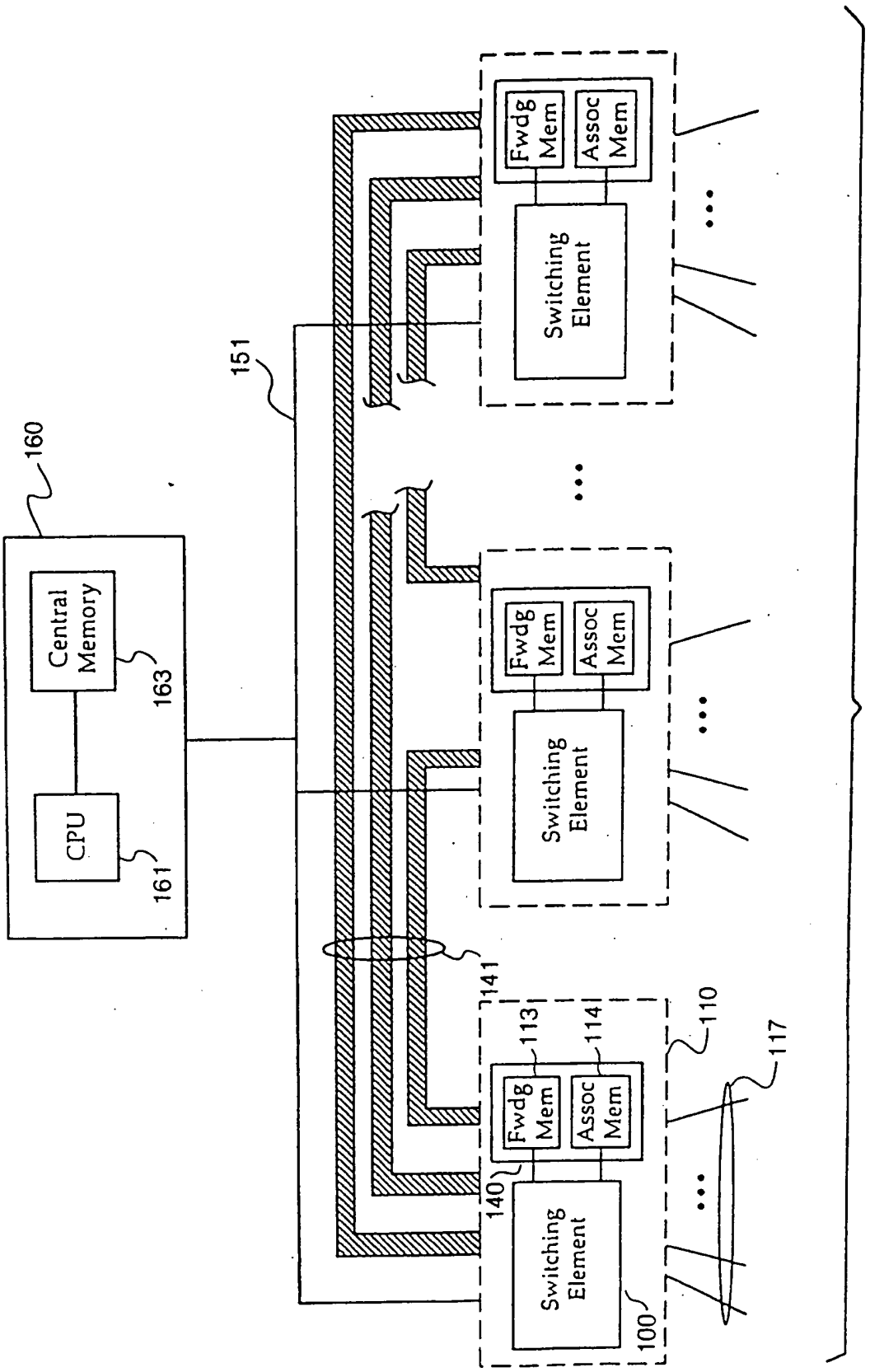

160
Central Memory
163
CPU
161
151
Fwdg Mem
Assoc Mem
Switching Element
141
113
114
110
117
140
100
101
To nodes and end-stations

FIG. 1

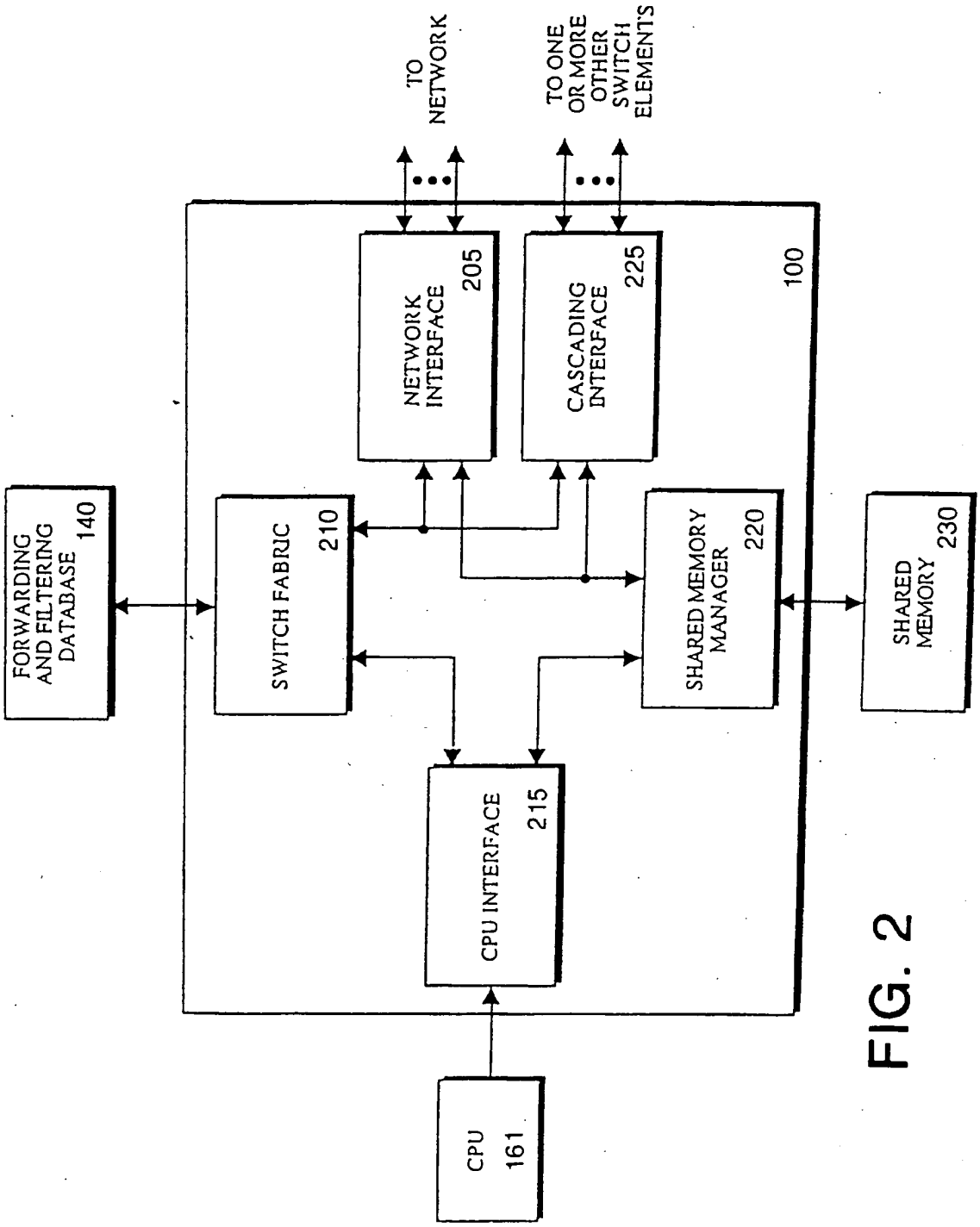

TO NETWORK
TO ONE OR MORE OTHER SWITCH ELEMENTS
NETWORK INTERFACE 205
CASCADING INTERFACE 225
100
FORWARDING AND FILTERING DATABASE 140
SWITCH FABRIC 210
SHARED MEMORY MANAGER 220
SHARED MEMORY 230
CPU INTERFACE 215
CPU 161

FIG. 2

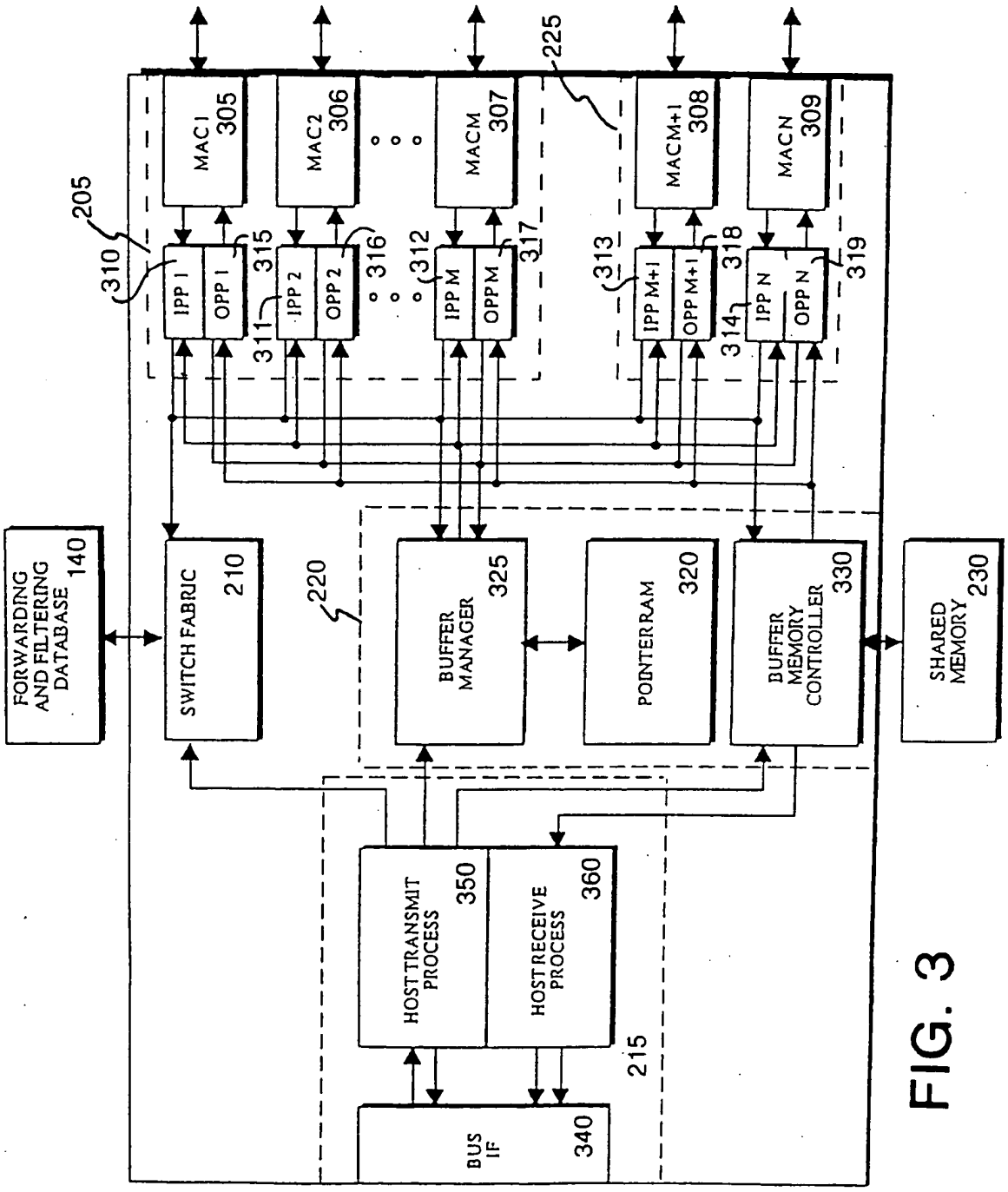

205
225
MAC1 305
MAC2 306
MACM 307
MACM+1 308
MACN 309
310
IPP 1
OPP 1
315
IPP 2
OPP 2
316
312
IPP M
OPP M
317
313
IPP M+1
OPP M+1
318
314
IPP N
OPP N
319
311
FORWARDING AND FILTERING DATABASE 140
SWITCH FABRIC 210
220
BUFFER MANAGER 325
POINTER RAM 320
BUFFER MEMORY CONTROLLER 330
SHARED MEMORY 230
HOST TRANSMIT PROCESS 350
HOST RECEIVE PROCESS 360
215
BUS IF 340

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/13199

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04L 12/28

US CL :Please See Extra Sheet.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 370/389, 428, 355, 356, 357, 359, 360, 362, 367, 372, 375, 379, 380, 328, 400, 402, 465, 468, 418, 434.

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X, P	US 5,724,358 A (HEADDRICK et al) 03 March 1998, see Figs. 1, 3, 5, and 7-10, col 3, lines 46-67, col. 4, lines 1-9, 19-28, and 35-41, col. 5, lines 43-54 and 66-67, col. 6, lines 1-17 and 61-67, col. 7, lines 1-67, col. 8, lines 1-67, and col. 9, lines 1-44.	1-6

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
27 AUGUST 1998	14 OCT 1998

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	PHIRIN SAM
Facsimile No. (703) 305-3230	Telephone No. (703) 308-9294

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/13199

A. CLASSIFICATION OF SUBJECT MATTER:
US CL :

370/389, 428, 355, 356, 357, 359, 360, 362, 367, 372, 375, 379, 380, 328, 400, 402, 465, 468, 418, 434.